

CalcOS Developer Manual

CalcOS reference

Date: 3/13/2024

Data codes

Calculators generate non-binary codes or data

Generating

On your calculator press 2 and -> two times, and press 2 again, now press √



Number	Byte	Data	Hex	
2	1.41E+11	0.001414		
5	2.24E+11	0.002236		
8	2.83E+11	0.028284		

This only works for common pocket calculators, it doesn't work on scientific calculator and in graphing calculators.

Note

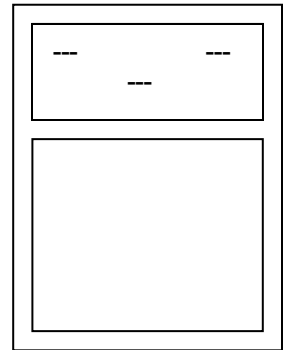
In this moment CalcOS is too limited, and only can show bytes and data

Developing

Blinking eyes example
main.mig

```
sys.set(engine."./prg.asm") {  
  plug.call([Calc]),  
  calc.display (  
    display.print[count 2, 0, _ = set.char];  
  ); else.fun() {  
    function blink() {  
      MeowType = (  
        const blink = display.print(count #)  
        var blink  
        blink.delay = 0.002ms  
      )  
    }  
  }  
};  
  
if(display.blink(0,1ms)) = (set "0" to "-" #);
```

Its only that for the moment



Understanding bytes

A byte is a unit of digital information that most commonly consists of eight bits. Historically, the byte was the number of bits used to encode a single character of text in a computer, and for this reason, it is the smallest addressable unit of memory in many computer architectures¹. Essentially, a byte represents a chunk of data, and it's often used to store or transmit information such as letters, numbers, or symbols. In binary terms, a byte can take on values from 0 to 255 (2^8 possibilities) because it's composed of eight bits, each of which can be either 0 or 1. So, when you see a file size expressed in kilobytes (KB), megabytes (MB), or gigabytes (GB), it's essentially a count of how many bytes are in that file

Calculator bytes

Normally calculators has 1 to 3 bytes of memory

Calculator memory map

	chunks	values			
Byte	1899	-99			
ROM	1000	-3			
CalcKter	998	-40			

CalcKter

CalcKter is the ROM of an pocket calculator (What make that works)

Its basically the Operating System of an calculator, but remember calculators doesn't run an full OS, but CalcOS make that real. CalcKter is an specialized software for performing calculations efficiently and is not too complicated, because we can see bytes on that, by creating an new memory section.

It uses assembly language to work.

Microprocessor

One example of such a microprocessor is the EMPCD081A.

It is commonly found in 8-digit calculators.

Features include basic arithmetic operations (addition, subtraction, multiplication, division), memory functions (M+, M-, MR, MC), and percentage calculations.

It operates with a clock frequency of 200 kHz during calculation and 18 kHz during wait operation.

The chip cannot shut down completely due to the need to refresh the LCD display and register key presses.

It draws a maximum of 20 μ A during operation.

The internal architecture likely involves a 4-bit logic for handling decimal calculations in base 10 (BCD format)



CalcOS assembly

Its not literary the assembly language but its MigScript with some plugins

- Calc
- MigAsm
- SysScaller

Code:

Please jump to the next page :3

```
sys.set(engine. "../prg.asm" ) {
  plug.call([Calc]),
  plug.call([MigAsm]),
  plug.call([SysScaller]),
  calc.MigAsm (
    function SysScaller() {
      MeowType() {
        keyboard(#01,#02,#03,#04,##complet.key())
        calc.display.panel(set.initial.char("0"))
        calc.display.panel(set.initial.msg("hello"))
        calc.mem({
          adress(asm.global) {
            keyboard."0x2f",
            display."0x3r",
            panel."0x044ef",
            mov: 10,
            mov: 20,
            mov: 30,
            mov: ##complet.dec
          }
          calc.micro(asm.global = SysScaller(4bit.1/2byte)) {
            call.micro(asm) {
              set.micro.clock(value = 0.11hz = [micro.make(
                micro.clock(set(var#))
              )))
            }
          }
        })
      }
    }
  )
}
```

Here we have an configuration file for our calculator, it uses global assembly (and it works on an 4-bit logic microprocessor), we are configuring the keyboard, display and panel, and the decimal code functions. And we have the processor clock configuration to 0.11Hz or 0.11 Bytes per second. The panel initial message is `hello` and the initial number is set to 0.

Configure calc-basic

What is BASIC?

BASIC (Beginners' All-purpose Symbolic Instruction Code)[1] is a family of general-purpose, high-level programming languages designed for ease of use. The original version was created by John G. Kemeny and Thomas E. Kurtz at Dartmouth College in 1963. They wanted to enable students in non-scientific fields to use computers. At the time, nearly all computers required writing custom software, which only scientists and mathematicians tended to learn.

In addition to the programming language, Kemeny and Kurtz developed the Dartmouth Time Sharing System (DTSS), which allowed multiple users to edit and run BASIC programs simultaneously on remote terminals. This general model became popular on minicomputer systems like the PDP-11 and Data General Nova in the late 1960s and early 1970s. Hewlett-Packard produced an entire computer line for this method of operation, introducing the HP2000 series in the late 1960s and continuing sales into the 1980s. Many early video games trace their history to one of these versions of BASIC.

The emergence of microcomputers in the mid-1970s led to the development of multiple BASIC dialects, including Microsoft BASIC in 1975. Due to the tiny main memory available on these machines, often 4 KB, a variety of Tiny BASIC dialects were also created. BASIC was available for almost any system of the era, and became the de facto programming language for home computer systems that emerged in the late 1970s. These PCs almost always had a BASIC interpreter installed by default, often in the machine's firmware or sometimes on a ROM cartridge.

BASIC declined in popularity in the 1990s, as more powerful microcomputers came to market and programming languages with advanced features (such as Pascal and C) became tenable on such computers. By then, most nontechnical personal computer users relied on pre-written applications rather than writing their own programs. In 1991, Microsoft released Visual Basic, combining an updated version of BASIC with a visual forms builder. This reigned use of the language and "VB" remains a major programming language[2][3] in the form of VB.NET, while a hobbyist scene for BASIC more broadly continues to exist.

Configuring assembly.mig file

Sysbasic.mig

```
sys.link(url("./assembly.mig"))
sys.set(engine."./prg.asm") {
  MeowType() {
    MigAsm.micro(const option = BASIC (value = ["2"]))
    calc.mem(
      adress(asm.global) {
        keyboard."0x2f",
        display."0x3r",
        panel."0x044ef",
        mov: 10,
        mov: 20,
        mov: 30,
        mov: ##complet.dec
      })
    MigAsm.micro.panel.display.key(
      export $BYTE
      function $BYTE() {
        sys.BYTE = (use.sys = equiv(bios, mem))
      }
      SysScaller.initial.set(ext = (".bas"), BASIC(sys.engine(start.session(code = "#"))))
      SysScaller.micro.panel(set.initial.msg("Calc BASIC" display.print($BYTE, $BASIC)))
    )
  }
}
```

It loads BASIC system and add the BASIC option on the AC menu

Datasette interface

The Commodore 1530 (C2N) Datasette, later also Datassette (a portmanteau of data and cassette), is Commodore's dedicated magnetic-tape data storage device. Using compact cassettes as the storage medium, it provides inexpensive storage to Commodore's 8-bit computers, including the PET, VIC-20, and Commodore 64. A physically similar model, Commodore 1531, was made for the Commodore 16 and Plus/4 series computers.

Features

The connection cable to the Datasette

The Datasette is probably the most sophisticated tape-storage method of any microcomputer.[1] Typical compact cassette interfaces of the late 1970s use a small controller in the computer to convert digital data to and from analog audio tones. The interface connects to the cassette deck using normal audio cables like RCA jacks or 3.5mm phone jacks. Such a system is compatible with the Apple II[2]: 57-58 and TRS-80 Color Computer, as well as many S-100 bus systems, and allows them to be used with any cassette player with suitable connections,[3] but using analog tones—similar to how dial-up modems work—often results in read errors instead of successful loading of data.[1]

The Datasette loading process

In the Datasette, instead of writing two tones to tape to indicate bits, patterns of square waves are used, including a parity bit. Programs are written twice to tape for error correction; if an error is detected when reading the first recording, the computer corrects it with data from the second.[1] The Datasette has built-in analog-to-digital converters and audio filters to convert the computer's digital data into analog audio and vice versa. Connection to the computer is done via a proprietary edge connector (Commodore 1530) or mini-DIN connector (1531). The absence of recordable audio signals on this interface makes the Datasette and clones the only cassette recorders usable with Commodore computers, until aftermarket converters made the use of ordinary recorders possible.

Because of its digital format the Datasette is both more reliable than other data cassette systems and very slow,[1][4] transferring data at around 50 bytes per second. After the Datasette's launch, however, special turbo tape software appeared, providing much faster loading and saving.[1] Such software was integrated into most commercial prerecorded applications (mostly games), as well as being available separately for loading and saving the users' homemade programs and data. These programs were only widely used in Europe, as the US market had long since moved onto disks.

Datasettes can typically store about 100 kByte per 30 minute side.[5] The use of turbo tape and other fast loaders increased this number to roughly 1000 kByte.

The Datasette has only one connection cable, with a 0.156-inch (4.0 mm)–spacing[6] PCB edge connector at the computer end. All input/output signals to the Datasette are all digital, and so all digital-to-analog conversion, and vice versa, is handled within the unit. Power is also included in this cable. The pinout is ground, +5 V DC, motor, read, write, key-sense.[7] The sense signal monitors the play, rewind, and fast-forward buttons but cannot differentiate between them. A mechanical interlock prevents any two of them from being pressed at the same time. The motor power is derived from the computer's unregulated 9 V DC supply [8] via a transistor circuit.[9]



BASIC

Using BASIC in a calculator can be complicated without a QWERY keyboard. But here is a solution, you can use the AUX port and plug in an AUX to usb adapter and connect an hub and a keyboard to it, and you can plug in a monitor to see advanced graphics on it

